FOR ARGUMENT

Supreme Court, U. S.
FILED
APR 6 1978
MICHAEL RODAK, JR., CLERK

IN THE

Supreme Court of the United States

October Term, 1977
No. 77-642

LUTRELLE F. PARKER, ACTING COMMISSIONER OF PATENTS AND TRADEMARKS,

Petitioner,

vs.

DALE R. FLOOK,

Respondent.

On Writ of Certiorari to the United States Court of Customs and Patent Appeals.

BRIEF AMICUS CURIAE FOR SOFTWARE ASSOCIATES, INC.

REED C. LAWLOR,
2500 East Colorado Blvd., Suite 350,
Pasadena, Calif. 91107,
Attorney for Amicus Curiae.

Of Counsel:
THEODORE H. LASSAGNE,
Hilton Center Office Building,
900 Wilshire Blvd., Suite 1000,
Los Angeles, Calif. 90017,

THOMAS N. WHITE, JR.,
3701 Highland Avenue, Suite 301,
Manhattan Beach, Calif. 90266.

Parker & Son, Inc., Law Printers, Los Angeles. Phone 724-6622

SUBJECT INDEX

Page

I.

The Questions 1

II.

Consent 2

III.

Interest of Software Associates, Inc. 2

IV.

General Argument 5

V.

Mathematical Equations and Algorithms Need to Be Distinguished and They May Involve Invention, Especially When Programmed Into a Machine 7

VI.

A Claim Should Not Be Rejected Merely Because It Involves Mathematical Relations or Algorithms 14

VII.

Benson's Claims Were Directed to a Mental Process and Not to a Process of Programming 18

VIII.

Programs Are Patentable Under the Present Patent Statutes 19

IX.

It Is Improper for the Petitioner to Dissect Flook's Claims and Then to Reject the Segments One Segment at a Time 20

X. Page

Important Computer Programs Are Not Freely Available at the Present Time but Are Sure to Become Freely Available Only if They Are Patented 24

XI.

The Availability of Copyright Protection for Computer Programs Does Not Justify Holding That Inventions That Utilize Machine Programs Are Nonstatutory 25

XII.

Conclusion 27

TABLE OF AUTHORITIES CITED

Cases Page

Atlantic & Pacific Tea Co. v. Supermarket Equipment Corp., 340 U.S. 147 (1950) 22

Chatfield, In re, 545 F. 2d 152 (1976), cert. den. 54 L.Ed. 2d 155 .....4, 22, 23

Dann v. Johnston, 425 U.S. 219 .....4, 18

Deutsch, In re, 553 F. 2d 21 (1977) 4

Faulkner v. Gibbs, 338 U.S. 267 .....15, 16, 17

Flook, In re, 559 F. 2d 21 (1977), cert. granted ..... F. 2d 4

Gottschalk v. Benson, 409 U.S. 63 (1972) 3, 4, 6, 7, 16, 17, 18, 19, 20

Joseph Schlitz Brewing Co. v. Houston Ice & Brewing Co., 250 U.S. 28 (1919) 22

MacKay Co. v. Radio Corp., 306 U.S. 86 (1939) 14, 16, 17

Noll, In re, 545 F. 2d 141 (1976), cert. den. 54 L.Ed. 2d 155 4

O'Reilly v. Morse, 15 How. 62 17

The Telephone Cases, 126 U.S. 534 (1887) 13, 17

Miscellaneous

Computerworld, Feb. 27, 1978 25

Report of the President's Commission on the Patent System, November 17, 1966 19

Senate Bill 1042, 89th Cong., 2nd Sess. 20

Rules **Page**

Rules of Practice of U.S. Patent and Trademark Office 1.58, 1976 (37 F.R. 21995, Oct. 18, 1972) 14

Statutes

Public Law 593, 82nd Cong., 2nd Sess. 19

United States Code, Title 17, Sec. 117 25

United States Code, Title 35, Sec. 101 .....5, 6, 23, 27

United States Code, Title 35, Sec. 102..... 6, 23, 26

United States Code, Title 35, Sec. 103 .....6, 22, 23, 26

Textbooks

Bender, David, "Trade Secret Protection of Software", 38 George Washington Law Review, July 1970, p. 909, reprinted in 3 Computer Law Service, Sec. 4-4, Art. 2 24

3 Computer Law Service, App. 3-2d, No. 3, p. 13 .. 25

3 Computer Law Service, App. 3-2b, No. 51, pp. 105, 106 25

Eames, Charles and Ray, A Computer Perspective, p. 33 (Harvard University Press, 1973) 9

Huskey, Henry D., et al., Computer Handbook, Sec. 16, "Digital-Computer-System Design" and Sec. 20, "General Purpose Computers" (McGraw-Hill Book Co., New York 1962) 11, 13

Knuth, Don E., Art of Computer Programming, Vol. 1, pp. 1ff (Addison-Wesley, Redding, PA, 1974) 8

Lawlor, Reed C., "Benson and Beyond", Computer Law Service, Sec. 4-2, Art. 3 (Callaghan, 1975), pp. 11-14 18

Page

Murphy, John S., Basics of Digital Computers (John F. Rider Publisher, Inc., 1958), pp. 76-84 11

Myers, Edith D., "Should Software be Copyrighted?", Datamation, Vol. 24, No. 3, March 1978, p. 132 .....24, 25

Ralston, Anthony, et al., Encyclopedia of Computer Sciences, pp. 47-48 (Petrocelli/Charter, New York 1976) 8

Schuyler, William E., Jr., "Protecting Property in Computer Software", reprinted from The Law of Software (George Washington University, 1969) in 3 Computer Law Service, Sec. 4-1, Art. 4 20

Stringham, Emerson, Patent Claims, A Drafter's Manual, Sec. 5513 (Pacot Publications, 1941) .. 14

IN THE

Supreme Court of the United States

October Term, 1977
No. 77-642

LUTRELLE F. PARKER, ACTING COMMISSIONER OF PATENTS AND TRADEMARKS,

Petitioner,

vs.

DALE R. FLOOK,

Respondent.

On Writ of Certiorari to the United States Court of Customs and Patent Appeals.

BRIEF AMICUS CURIAE FOR SOFTWARE ASSOCIATES, INC.

I.
The Questions.

Three principal questions are presented by this case.

Question No. 1

Is Flook's claimed invention to a mathematical equation or is it to a petroleum refining process which has been modified in accordance with a mathematical equation?

Question No. 2

Is a claim nonstatutory if the mathematical aspects of a relationship involved in the claim may be carried out by means of a computer?

Question No. 3

Is programming of a general purpose digital computer in accordance with a mathematical algorithm always unpatentable?

The first question is not faced by either the Petitioner or the Amicus CBEMA.[1] They both assume an answer to the first question, believed to be incorrect by Amicus. The second question is raised indirectly in the briefs of both the Petitioner and the CBEMA. The third question is touched upon lightly by the Petitioner, but answered in the negative by CBEMA.

II.
Consent.

Software Associates, Inc. has received consent from both parties to file this brief amicus curiae. The letters of consent have been filed with the Clerk of the Court.

III.
Interest of Software Associates, Inc.

This Amicus, Software Associates, Inc., is a small corporation located in Altadena, California and engaged in the manufacture of software for a wide variety of customers. Amicus believes that it is typical of thousands of other small companies engaged in software manufacture.

Amicus has no interest in Flook's patent application and does not consider itself prepared to form a judgment as to whether or not the invention described and claimed there is unobvious over the prior art.

---

[1]The following abbreviations are employed in this brief: "Pet. Br." refers to brief of the Petitioner; "CBEMA" refers to the Computer and Business Equipment Manufacturers Association; "CBEMA Br." refers to the brief of the CBEMA.

However, Amicus believes that it is qualified to be of assistance to the Court in its consideration of mathematical equations, mathematical algorithms, computer algorithms, and the manner in which they may be related to other parts of mechanical, electrical, and chemical devices and systems.

Because of the uncertain status of the patent law, at least as it appears to the software industry, Amicus has not sought to protect any of its products under the patent laws. Discouraged, Amicus, like many other similar companies, relies upon the law of trade secrets and the law of contracts in order to protect whatever proprietary rights that it has.

Amicus is familiar with the benefits that small companies derive from the United States patent laws in other fields. By obtaining patents on their inventions, small companies in other industries are given an opportunity to grow, and sometimes do succeed in growing. Amicus believes that companies engaged in software manufacture deserve to benefit from the patent laws to the same extent as companies in other industries, when they make meritorious inventions.

Amicus is familiar with *Gottschalk v. Benson*, 409 U.S. 63 (1972) and is also familiar with many of the practices of the United States Patent and Trademark Office. Amicus is confused by the interpretation that the Petitioner and CBEMA have placed on *Gottschalk v. Benson, supra,* and is dismayed by the prospective costs and difficulties of obtaining a patent on an invention that involves computer programming and especially by the prospects that a patent may not be obtained at all without the inventor being forced to bring the application up to the Court of Customs and Patent

Appeals and even all the way up to this Court for review. This is especially so since 5 cases have been decided by the Court of Customs and Patent Appeals in favor of patent applicants since the decision in *Gottschalk v. Benson, supra,* was rendered by this Court, and of these 5 cases, the Petitioner has sought to have the decisions of the Court of Customs and Patent Appeals set aside by this Court in 4 of them.[2] The only exception is *In re Deutsch,* 553 F. 2d 21 (1977).

Amicus recognizes that the Petitioner has stated the issue in this case in terms of a mathematical equation but Petitioner has also indirectly raised the issue as to whether any invention at all would be patentable if in the course of practicing the invention a mathematical equation is programmed into a computer and the programmed computer is used as part of a larger system.[3]

Amicus submits that the programming of mathematical equations or mathematical algorithms into a computer to form machine algorithms can so alter the mode of operation of the computer or of the system in which it is used, that the combination that involves such programming of a computer may be sufficiently meritorious to deserve patent protection and that any contrary conclusions would be injurious to the public welfare.

---

[2]*In re Noll,* 545 F. 2d 141 (1976); cert. den. 54 L.Ed. 2d 155. *In re Chatfield,* 545 F. 2d 152 (1976), cert. den. 54 L.Ed. 2d 155. *In re Flook,* 559 F. 2d 21 (1977), cert. granted ..... F. 2d ..... *Dann v. Johnston,* 425 U.S. 219.

[3]Pet. Br., p. 23.

IV.

General Argument.

It is the view of Amicus that Question No. 1 should be answered in favor of the proposition that Flook's claim (*infra,* pp. 20-21) is for a petroleum refining process which has been modified in accordance with a mathematical equation, thus rendering Flook's claim proper under 35 U.S.C. 101.

The Petitioner and the CBEMA have not faced up to Question No. 1 but have asserted, without proof, that Flook's invention resides *solely* in the mathematical equation that he employs for modifying the last step of his process.

Flook claims a new method of regulating a system involving the catalytic chemical conversion of hydrocarbons in which the last step is related in a new way to conditions detected in the performance of the process. The new relation, which is set forth in the last step, is related to observations made of the process as expressed in the form of a mathematical equation. But this equation is neither a mathematical algorithm nor a machine, or computer, algorithm.

Both Petitioner and CBEMA seem to concede that the Flook claims define statutory subject matter if the invention is in the petroleum processing method and not solely in the mathematical equation employed by Flook in his process.

So the real issue in this case is one of fact. It is submitted that this fact issue should be resolved in favor of Respondent.

Even if the decision of this Court is to the contrary with respect to Question No. 1, it is the position

of Amicus that decisions on Questions No. 2 and No. 3 should be in the negative.

Furthermore, the arguments of Petitioner and the CBEMA are somewhat confusing and misleading because they have not distinguished the meanings of important terms.

Part of the difficulty resides in the fact that various people, including particularly the CBEMA, often disregard distinctions between a mathematical algorithm and a machine, or computer, algorithm. Both the Petitioner and the CBEMA, especially the former, also fail to distinguish an algorithm from a mathematical equation. The meanings which they attribute to these words appears to shift throughout their arguments, and they refer to various parts of the *Benson* case without taking these important distinctions into account.

In an effort to clarify some of the problems presented by this case, this brief will emphasize the facts that are involved in the various cases discussed, both this case and in precedent cases, so as to bring out the relationship between the *claimed* subject matter of this case and the *claimed* subject matters of prior cases.

Petitioner has dissected Flook's claims and has found fault with them part by part. Such dissection is not permissible under 35 U.S.C. 102 or under 35 U.S.C. 103 and should not be permitted under 35 U.S.C. 101.

In any event, contrary to the apparent assumptions of Petitioner and the CBEMA, programs, at least machine programs, constitute patentable subject matter under the present patent statute.

Under all the authorities taken as a whole, a claim should not be held to be nonstatutory merely because it involves relations that are expressed by a mathematical equation or by an algorithm of any kind.

V.

Mathematical Equations and Algorithms Need to Be Distinguished and They May Involve Invention, Especially When Programmed Into a Machine.

It is believed that it will be helpful in understanding the present invention, the *Benson* case, and a number of the precedent cases cited hereinafter, if the terms "mathematical equation", "mathematical algorithm", and "computer", or "machine", "algorithm" are distinguished.[4]

It was correctly recognized by this Court in *Gottschalk v. Benson, supra,* that an algorithm is a procedure for solving an equation. In other words, a mathematical equation is not an algorithm. This important distinction is not taken into account by either the Petitioner or the CBEMA.

A mathematical equation may express relations between values that represent physical phenomena. As will be seen, these relations are commonly of two kinds, algebraic, or quantitative, relations and logical relations.[5] Both types of mathematical relations are important in computer programming technology.

---

[4]For simplicity we use the term "machine algorithm" to refer to a program that is stored in a machine and which causes the machine to carry out a sequence of operations when the machine operates. And we use the term "mathematical algorithm" to refer to a procedure for solving a mathematical problem with the aid of pencil and paper or just in one's head.

[5]The mathematics of logical relations is involved in "modern math".

A procedure for solving an equation may be a pencil and paper, i.e., a mental, process or it may be a mechanical or electrical process. In the field of modern computer technology, an algorithm that operates within a computer involves a step-by-step series of changes of states of electrical or magnetic elements. Programming a computer to solve a mathematical equation also involves changing states of such physical elements.

The term "algorithm" is used in the computer art almost as a synonym for any kind of a program in or for a computer. And sometimes a mathematical equation is called an algorithm. Hence, there is a need to use the term "algorithm" with caution. It might be a printed program for making calculations step-by-step, in accordance with a mathematical algorithm. It may also be a printed listing of a series of instructions that are to be programmed into a computer to alter its internal structure and functions by changing states of magnetic or electrical elements. Such a list of instructions is sometimes known as software. (But software is a generic term with additional meanings.) But it might also be in machine form within a computer to cause the computer to perform a series of steps in a predetermined sequence.[6] The actual sequence that is followed may vary from time to time depending upon the values of the variables supplied to the computer.

---

[6]Knuth, Don E., *Art of Computer Programming*, Vol. 1, pp. 1 ff (Addison-Wesley, Redding PA, 1974); see also Ralston, Anthony, et al., *Encyclopedia of Computer Sciences*, pp. 47-48 (Petrocelli/Charter, New York 1976).

Computer algorithms make use of logical equations as well as ordinary algebraic equations and the various parts of a computer that are affected by algorithms are often referred to as logic elements.

The equation

$$V = LWH \qquad (1)$$

and the equation

$$L = S_1 S_2 S_3 \qquad (2)$$

represent the two forms of the aforementioned equations. Equations (1) and (2) are both mathematical equations of the kind to which the statements of the Petitioner and the CBEMA seem to refer. Though equations (1) and (2) are identical in form, they may represent entirely different physical concepts. They may even be subject to entirely different mathematical laws. Thus, equation (1) may represent an algebraic or quantitative relation, whereas equation (2) may be a logical relation. We will illustrate the difference.

In the field of ordinary algebra, equation (1) may be employed to represent the rule that the volume (V) of a box is obtained by multiplying its length (L) by its width (W) and then multiplying the resultant product by the height (H) of the box.

But in the field of Boolean algebra, which is a branch of symbolic logic, the second equation may represent the proposition that a signal circuit (L) for turning a light on and off is closed only when all three switches S1, S2, and S3 in series are closed.[7]

---

[7]The underlying concept was first suggested by Charles S. Pierce, the founder of the philosophy of pragmatism, in a letter written in 1879 (see Eames, Charles and Ray, *A Computer Perspective*, p. 33 (Harvard University Press, 1973)).

Equation (2) is now called a logical product, and the circuit in which it is connected is sometimes referred to as an AND gate. Modern computers use AND gates and other somewhat similar devices, such as OR gates and NOT units, but of a somewhat more sophisticated kind. But they are still essentially switching circuits.

A modern binary digital computer consists of many logic units, such as AND gates, OR gates, and NOT units and of many subassemblies, such as storage registers, shift registers, counters, coders, decoders, and the like, which themselves are made up of such elementary logic units. They also include memory units and a sequence controller.

In such a computer, certain memory units are employed for storing programs temporarily by means of "software" operations and other memory units called "read only memory" (ROM) units have sections of programs stored in them, often permanently. (Some ROM units are also reprogrammable under special conditions.) The former involves "software" programs. The latter involves "firmware" programs. The terms refer in part to the ease with which a program may be changed in the machine.

When such a digital computer is programmed, machine instructions are created in a memory unit by software. Such instructions are retained in the memory unit for later use when selected by the sequencer. When the program operates, the sequencer fetches these instructions one at a time. When each instruction is fetched, it causes an operation to be performed, such as a "logical product" operation or an arithmetic "add" operation. Normally, many of the instructions cause

various parts of the computer to operate in accordance with logic equations even when the computer is programmed to solve an ordinary algebraic equation.

The machine instructions are stored in a programmed sequence, such that when the sequencer fetches them, various operations occur in a predetermined sequence depending upon the conditions that exist in the computer at that time as a result of input data and prior operations caused by carrying out steps in the algorithm. In effect, each instruction causes various machine elements to be enabled or disabled to carry out the operation desired.[8] In programming a computer to solve a mathematical equation, a machine algorithm is established in a memory unit to carry out a procedure in which the computer performs a sequence of operations one step at a time and then finally ends up with a solution to the problem. But computers do not only solve mathematical problems. Quite usually the program is set up to relate the computer and some of its operations to external parts of a larger system in a special way and to cause various parts of the larger system to be interconnected and to operate upon each other in a manner determined by the machine algorithm.

Each time such a computer is programmed, the set of instructions previously stored in the computer is, in effect, erased or destroyed and a new set of instructions is stored, thus enabling the computer to function in a different way when operating with the new program than it did with the prior program.

---

[8]Huskey, Henry D., et al., *Computer Handbook*, Sec. 16 "Digital-Computer-System Design" and Sec. 20 "General Purpose Computers" (McGraw-Hill Book Co., New York 1962); Murphy, John S., Basics of Digital Computers (John F. Rider Publisher, Inc., 1958), pp. 76-84.

In actual practice, in modern binary digital computer technology, a mathematical equation can be fed into a computer from punched cards, from a remote typewriter-like keyboard, or from a coded magnetic tape or disc, to construct a computer algorithm that carries out a procedure for solving the equation. Normally, when this is done, the equation fed in is processed by means of what is known as a "compiler" and translated into the machine, or computer, algorithm. This machine algorithm is usually generated automatically by the compiler for solving the equation. So long as this is all that is done, it could be argued that the inventor has made no patentable invention.

But the inventor often does more. This could occur, for example, if the computer is employed to regulate a petroleum refining process in a new way, as in Flook's case. It could also occur if the compiler or computer or both, function in a new and unobvious way. It could also occur if the inventor develops a new and unobvious kind of compiler or a new and unobvious kind of memory bank or a new data processing unit. It could also occur where the inventor first converts his mathematical equation into a partial mathematical algorithm and then feeds the mathematical algorithm through a compiler to the computer to create a new and unobvious machine algorithm which solves the original mathematical equation in a particularly efficient, or reliable, way. The possibilities are endless. The Petitioner seems to urge that none of these can constitute statutory subject matter.

The system of programming described above involves what is called a stored program. In fact, we have emphasized programs that are only temporarily stored. As mentioned above, some programs and parts of programs are stored permanently in magnetic or electrical elements. Programming, however, may also be accomplished with the aid of plug boards and with permanent wiring.[9] In a computer of the stored-program type, in effect, the program is created by changing the states of memory elements, and the program is carried out by changing the states of other elements to provide, in effect, electrical connections between various computer elements to carry out the various operations called for by the stored-program.

The actual machine algorithm created by software programming of a computer depends upon the construction of the machine and the nature of the programming method, and in the case of stored programs, upon the nature of the compiler. The key though to the determination of the principal related computer function is in the mathematical equation itself, or in the particular mathematical algorithm for establishing a particularly efficient procedure for solving the equation.

In ordinary operations, the mechanical arrangement of the various elements of the computer remains fixed, even though the states of these elements are altered by programming and even though the states of the elements are altered sequentially during operations. It is well established that invention may be involved in a system in which changing electric currents represent information. *The Telephone Cases,* 126 U.S. 534 (1887).

No one could possibly know at the present time all the ways that the action of the general purpose binary digital computer could perform with the programs that have not yet been invented. Hence, it would

---

[9]Huskey, et al., see Footnote 8.

be dangerous to deny the benefits of the patent laws to all inventions that involve mathematical relations or programs.

VI.

A Claim Should Not Be Rejected Merely Because It Involves Mathematical Relations or Algorithms.

It has long been recognized that mathematical terms may be employed for helping to define an invention in the claims.[10] And this Court has sustained claims that involved a mathematical equation, either expressed or implied, to help define an invention, regardless of the novelty of the equation.

An apparatus claim may be defined by reference to relations between parts. A process claim may be expressed in terms of steps, conditions under which the steps are carried out, and how the steps are related to each other. The relations may be expressed by mathematical equations.

In *MacKay Co. v. Radio Corp.*, 306 U.S. 86 (1939), this Court upheld the decision of a lower court holding claims to a radio antenna valid when the relative positions of old parts of the antenna were defined in terms of an old mathematical equation. Clearly, the patent was for an antenna, not for a mathematical equation per se. Under the Petitioner's view, the claim would be nonstatutory since, he would argue, the "sole novelty of the claim consists of the mathematical equation". Actually, the invention was in the relationship of the parts, which relationship could best be expressed by a mathematical equation.

---

[10]Stringham, Emerson, *Patent Claims, A Drafter's Manual*, Section 5513 (Pacot Publications, 1941); Rules of Practice of U.S. Patent and Trademark Office 1.58, 1976 (37 F.R. 21995, Oct. 18, 1972).

In *Faulkner v. Gibbs*, 338 U.S. 267, claims to a game were held valid. The claims involved electrical circuitry that solved an unexpressed mathematical equation.[11] The *Gibbs* patent claimed a marble game machine, the kind that had a spring switch beneath each of 25 holes on a sloping board, into which marbles were shot one at a time. The holes were arranged in a 5x5 square pattern in the shape of a BINGO card.

In this particular case, the machine solved a number of mathematical equations. A typical equation that was solved by the machine involved a logical product of the kind mentioned above, except that there were five switches that had to be closed for a particular light circuit to close. Such a mathematical equation would be of the form

$$L = S_1 \cdot S_2 \, S_3 \, S_4 \, S_5$$

---

[11]Claim 9 of Gibbs Patent No. 1,906,260 held valid by the Court below (170 F. 2d 34, CA9) reads as follows:

> "9. A game apparatus comprising a plurality of electrically connected units, each including a game board with a plurality of apertures therein, and an annunciator with a plurality of indicators thereon, electrical contacts adjacent each of said apertures connected in the circuits of said indicators, said indicators and said apertures corresponding in number and subdivided into corresponding groups, whereby when objects are deposited in said apertures by the players at the several units, corresponding indicators will be energized, a supplementary signal circuit on each of said units, and *AND means for holding said signal circuit open until all of the indicators of any group on each of said units have been energized, and for closing said signal circuit when all of the indicators of any unit have been energized,* and means controlled by the closing of the signal circuit of the winning unit for discontinuing the signals and opening the circuits of the indicators on all other units." [Italicizing and capitalized word AND have been added for clarity.]

The italicized portion sets forth a logical product of the type of equation (2) set forth above.

A number of such equations could be used to describe the function of an ordinary game board of the BINGO type. When the problem was solved, lights remained operated to indicate that balls had entered five holes in a row of the 5x5 array. A number of game boards were operated simultaneously by competing players and NOT units opened circuits of the losers when an AND circuit of the winner was closed.

In the *MacKay* case, the equation was of the ordinary algebraic type. In the *Faulkner* case, no mathematical expression was set forth, but the operation could be described in terms of a mathematical relation of the kind known as a logical product. Over 2000 years ago, Pythagoras taught that all phenomena could be described in mathematical terms. In both the *MacKay* case and in the *Faulkner* case, the claims would have been pulverized to dust if the claims had not been considered as a whole and held together by the mathematical relations as limitations, whether expressed in formula form or whether described in words.

The claims of the *MacKay* case provide a perfect example of what would happen if the mathematical equation set forth in a claim could be disregarded. Nevertheless, the validity of the claim was upheld. Clearly, the reason for this is that combining the old parts in accordance with the old equation created a new and unobvious structure in which the old parts were interrelated in a new way.

The *Benson* case did not overrule *MacKay* or *Faulkner* but applied only to a set of claims which could be infringed without the use of any apparatus at all (*Gottschalk v. Benson*, 409 U.S. 63 (1972)).

The *Benson* case helped draw a line between a claim to a mathematical algorithm for solving an equation and a claim to a combination of a plurality of physical elements that were expressly, or inherently, interrelated in a mathematical manner.

Accordingly, these three cases of this Court taken as a whole must mean that a claim to a process is not nonstatutory merely because a step of a claim is carried out in accordance with an equation or with the aid of either a mathematical algorithm or a machine algorithm.

It is reasonable to apply the rules of *MacKay* and *Faulkner* to process claims.

At page 12 of his brief, the Petitioner treats mathematical formulae and algorithms the same as abstract ideas and scientific principles and phenomena of nature. It is well established that a claim constitutes statutory subject matter even if it involves an invention that is an application of a mathematical equation, an abstract idea, a scientific principle, or a law of nature. *MacKay Co. v. Radio Corp., supra; Faulkner v. Gibbs, supra; O'Reilly v. Morse*, 15 How. 62; *The Telephone Cases, supra; Gottschalk v. Benson, supra.*

Accordingly, the decision of the CCPA is consistent with the law of all prior cases of this Court.

VII.

Benson's Claims Were Directed to a Mental Process and Not to a Process of Programming.

The *Benson* case is to be construed narrowly, not broadly in the manner sought by Petitioner and the CBEMA. *Dann v. Johnston,* 425 U.S. 219, 224.

In holding Benson's claims nonstatutory, the Court pointed out that the *claimed* invention "could be performed *without any apparatus*" at all. In other words, the *Benson* case was concerned only with whether a particular, simple mathematical algorithm (not a machine algorithm) was statutory subject matter.

This conclusion can be understood only if it is recognized that the term "re-entrant register" used in Benson's claim 8, was non-apparatus. This indeed was the case[12], and this was recognized by this Court and distilled in the simple statement quoted above. As explained more fully by this Court, Benson's claims were not limited to any particular art or technology, to any particular apparatus or machinery, or to any particular end use, and the claimed process did not operate to change articles or materials to a "different state or thing". In other words, Benson's claims could be infringed by a mental process alone.

A study of the *Benson* case therefore makes it clear that the *Benson* case was not concerned with programming of a computer or with the resultant programmed computer and certainly not with any relation of a mathematical equation or of a programmed computer to other parts of a claimed process or apparatus.

---

[12]Lawlor, Reed C., "Benson and Beyond", *Computer Law Service,* Sec. 4-2, Art. 3 (Callaghan, 1975), pp. 11-14.

VIII.

Programs Are Patentable Under the Present Patent Statutes.

The Petitioner seems to urge (Pet. Br. p. 23) that "the patenting of computer applications of mathematical algorithms" is not permissible under present law. The CBEMA seems to make the same assumption, at least up to a point (CBEMA Br. pp. 4, 15-16). Their viewpoints are based in part upon a misinterpretation of the present patent statutes, and the Report of the President's Commission, *infra,* and the *Benson* case, *supra.*

It is the position of Amicus that computer programs, though not mentioned, *in haec verba,* in the statute, clearly constitute patentable subject matter under the present patent law.[13]

That programs are patentable under present law was not denied by the President's Commission on the Patent System in 1967, and sharply pointed out by William E. Schuyler, the Commissioner of Patents in 1969.

In its report,[14] the President's Commission proposed that a *new* patent law be enacted in which

> "The classes of patentable subject matter *shall continue as at present, except*
>
> . . .
>
> A series of instructions which control or condition the operation of a data processing machine, generally referred to as a 'program' shall not be considered patentable . . ." [Italics added.]

---

[13]Public Law 593-82nd Congress, 2nd Session; United States Code, Title 35.

[14]Report of the President's Commission on the Patent System, November 17, 1966.

William E. Schuyler, while he was Commissioner of Patents, summarized it this way,

> "The President's Commission to study the patent system recommended that computer programs be *excised* from the patent system". [Italics added.][15]

Attempts to change the law (*e.g.*, S1042, 89th Congress, 2nd Session) in order to excise computer programs from the classes of statutory subject matter have been rejected by Congressional Committees. It is therefore believed to be clear that computer programs are patentable under current law.

However, perhaps, as recommended by this Court in *Benson*, Congressional action is needed if statutory protection is to be extended to purely mathematical algorithms of the kind that are not limited "to any particular art or technology, to any particular apparatus or machinery, or to any particular end use", and which are "so abstract and sweeping . . . [that] the end use may . . . be performed . . . without any apparatus" at all.

IX.

It Is Improper for the Petitioner to Dissect Flook's Claims and Then to Reject the Segments One Segment at a Time.

Flook's claim 1 reads as follows:

> "1. A method for updating the value of at least one alarm limit on at least one process variable involved in a process comprising the catalytic

[15]Schuyler, William E., Jr., "Protecting Property in Computer Software", reprinted from *The Law of Software* (George Washington University, 1969) in 3 *Computer Law Service*, Sec. 4-1, Art. 4.

> chemical conversion of hydrocarbons wherein said alarm limit has a current value of
>
> Bo + K
>
> wherein Bo is the current alarm base and K is a predetermined alarm offset which comprises:
>
> (1) determining the present value of said process variable, said present value being defined as PVL:
>
> (2) determining a new alarm base B1 using the following equation:
>
> B1 = Bo(1.0-F) + PVL(F)
>
> where F is a predetermined number greater than zero and less than 1.0;
>
> (3) determining an updated alarm limit value which is defined as
>
> B1 + K; and, thereafter
>
> (4) adjusting said alarm limit *to said updated alarm limit value*." [Italicizing added.]

The Petitioner argues that there is no novelty in the refining process, but only in the mathematical equation. In this he ignores the novel relationship established between the steps of the process.

What Petitioner does, is dissect the last step (4) of the Flook claim into two portions, one of which is underlined above and the other of which is not underlined. In considering the novelty of Flook's contribution, Petitioner then ignores the underlined portion because the updated value referred to there was calculated by use of a mathematical formula referred to earlier in the claim, and then he declares that the remainder of that last element is not new, and hence that the entire claim is old. At the very best,

that is an untenable, unsupported rejection under 35 U.S.C. 103.

Dissection of any step of a claim amounts to dissection of the whole claim. Such dissection is forbidden by 35 U.S.C. 103. For it is the claim as a whole which must be considered, not the individual parts (35 U.S.C. 103).

Most, if not all, claims refer to old parts, old properties, and old relations. If a claim is to be declared inappropriate for consideration under the patent law because there is some objection to each of the individual parts separately, all patents and the patent law itself would be destroyed one part at a time.

Such destruction would not be limited to mathematical relations or mathematical algorithms, or computer algorithms. It would cut across all other technological arts as well.

A claim represents a bundle of features, namely, parts, steps, properties, relations, conditions, and the like. The strength of the bundle is greater than the strength of the individual parts and hence may be patenable, especially in an electrical or chemical case, *Atlantic & Pacific Tea Co. v. Supermarket Equipment Corp.*, 340 U.S. 147 (1950). What Petitioner does is seek to destroy the fagot, stick by stick. Just as this practice has been condemned in the trademark law, *Joseph Schlitz Brewing Co. v. Houston Ice & Brewing Co.*, 250 U.S. 28 (1919), it should also be condemned in the patent law.

The Court of Customs and Patent Appeals has expressed its views on this very point in the case of program-related inventions, in *In re Chatfield, supra,* in these words:

> "Our references in Christensen to the mathematical equation as being 'at the point of novelty' does not equate to a holding that a claim may be dissected, the claim components searched in the prior art, and, if the only component found novel is outside the statutory classes of invention, the claim may be rejected under 35 USC 101. That procedure is neither correct nor within the intent of Congress, for the reasons we stated in Bernhart."

The Petitioner, in dissecting the claim into separate sticks, urges that a patent on the fagot would be a patent on each stick separately, and then jumps to the illogical conclusion that since a mathematical equation is one of the sticks of Flook's claim, a patent on the fagot would be a patent on the mathematical equation.

The Petitioner seeks to destroy the bundle represented by a claim by destroying one feature at a time. Such practice is not permitted under 35 U.S.C. 102 or under 35 U.S.C. 103. And it should not be permitted under 35 U.S.C. 101.

Dissection of a claim into sticks to destroy the fagot should not be condoned by this Court.

Flook does not seek a patent on the mathematical equation but only on his method of setting alarm limits for a chemical process by adjusting those limits in accordance with a determination made with the aid of a mathematical equation and prior determination of the state of the process.

X.

Important Computer Programs Are Not Freely Available at the Present Time but Are Sure to Become Freely Available Only if They Are Patented.

CBEMA strongly implies[16] that there has been a free interchange of computer programs throughout the history of the computer industry. This is simply not so with respect to important programs.

At the present time, important programs are guarded in secrecy, largely because the Petitioner refuses to recognize their merit from a patent standpoint.

One author puts it this way,

> "Trade secret law is by far the most widely used method of attempting to secure legal protection for software".[17]

Reliance on trade secret protection has continued to grow since that article was written. And lawyers even recommend that clients who mark their programs with statutory copyright notices, also treat the programs as trade secrets.[18] The notice preserves copyright if and when the secret is lost.

If programs are patented, then they will become freely available to the general public after the expiration of the 17-year grant provided under the patent law. If they are published and covered by copyright, they will become freely available after about 75 years. But since neither patent protection not copyright protection

---

[16]CBEMA Br. 17-21.

[17]Bender, David, "Trade Secret Protection of Software", *38 George Washington Law Review*, July 1970, p. 909, reprinted in 3 *Computer Law Service*, Sec. 4-4, Art. 2.

[18]Myers, Edith D., "Should Software be Copyrighted?", *Datamation*, Vol. 24, No. 3, March 1978.

is considered adequate by American business at the present time, large numbers of important programs are maintained in secrecy, Manufacturers of computers (without doubt including many computer-manufacturer members of CBEMA) are particularly zealous to preserve their programs in secrecy[19] and do not make them available on a free interchange basis.[20]

XI.

The Availability of Copyright Protection for Computer Programs Does Not Justify Holding That Inventions That Utilize Machine Programs Are Non-statutory.

CBEMA states that computer programs are copyrightable. Amicus agrees. Though no court has passed on this question, program listings are believed to be copyrightable under the new law, as explained by CBEMA, as well as under the old law. However, other provisions of the new Copyright Act, which became effective January 1, 1978, are in a state of suspension (17 U.S.C. 117), and it has been reported that CONTU plans to recommend against extending the benefits of the new law to computer programs.[21] It is implied by CBEMA that since copyright protection is available

---

[19]A typical agreement reads in part as follows: ". . . All programs furnished by . . . shall remain confidential and proprietary property of . . . Customer agrees to continue to treat such information as confidential and proprietary property of . . . and shall acquire no rights in them except to use such information solely for the purpose of and only during the time it uses the equipment or for the period covered by any license fee. Etc.", 3 *Computer Law Service*, App. 3-2d, No. 3, p. 13.

[20]See the foregoing agreement and another at 3 *Computer Law Service*, App. 3-2b, No. 51, pp. 105, 106.

[21]*Computerworld*, Feb. 27, 1978; See also, Myers, Note 18, *supra*, p. 132.

for computer programs, patent protection should not be.[22] Amicus cannot agree with this conclusion.

To argue that computer programs should not be patenable because copyright protection is available for program writings as works of authorship would be like arguing that computer circuits should not be patentable because drawings of computer circuits are copyrightable.

The requirements for copyrightability and for patentability are entirely different, the terms are entirely different, and the rights established by issuance of a certificate of copyright and by issuance of a patent are entirely different.

Copyrights are obtainable for programs that are not new so long as they involve originality. Patents should be obtainable for inventions that involve machine programs if they meet the usual standards of novelty (35 U.S.C. 102) and non-obviousness (35 U.S.C. 103).

The parts of a general purpose digital computer are interrelated in a new way when it is reprogrammed. In some respects, a general purpose digital computer is like a toy of a kind with which children play, such as an Erector set, a Mechano set, or a Tinker Toy set, and that can be employed for constructing various objects. Patents issue on such sets, and parts thereof, and instructions are usually supplied to build various devices with such construction sets. A person could write his own set of instructions for constructing a new device. The written instructions would be copyrightable, just like a program listing; and if published without copyrighting, they could be freely copied by others. But the device created by following the instructions could very well be patentable either alone or

[22]CBEMA Br. pp. 5, 26-27.

in combination with another device. Kits, similar to such mechanical sets, are also available for creating new electronic circuits and even computers or parts thereof. Clearly, a device constructed with such a set or kit would constitute statutory subject matter under 35 U.S.C. 101, regardless of the copyrightability of the written instructions. A modern general purpose binary digital computer is like a kit in which the coaction of the various elements and subassemblies with each other and with external components are changed by software programming. A patent based upon that new creation would not be a patent on the written instructions.

Patent protection should not be denied to creators of program-related inventions merely because programs may be copyrightable when in written form.

XII.
Conclusion.

For the reasons set forth above, Amicus urges this Court to hold that an invention that is described by means of a mathematical equation or a mathematical algorithm and which may be practiced with the aid of a computer programmed in part by such equation or such mathematical algorithm, constitutes statutory subject matter under 35 U.S.C. 101. Such a holding would be consistent with all of this Court's prior decisions.

Respectfully submitted,

REED C. LAWLOR,

Attorney for Amicus Curiae.

Of Counsel:

THEODORE H. LASSAGNE,

THOMAS N. WHITE, JR.,